COMP90049 Report:
Comparing The Performance of Naïve Bayes, Decision Tree, and Multi-Layer Perceptron On Movie Classification

Anonymous

1 Introduction

Nowadays, many different machine learning algorithms have been developed and applied into real world cases. However, different models have distinct properties and performance which might lead to quite different results. Thus, choosing the suitable model is a critical issue for engineers. And this report compared the performance of three different machine learing models on a movie classification test.

In this report, the author first reviewed some widely applied machine learning models. And then, selected three models such as Naïve Bayes, Decision Tree, and Multi-Layer Perceptron to implenment in the movie classification test with relevant dataset [1][2]. This dataset includes both numerical and vocabulary data. Thus, the vocabulary data were converted into numerical type using one-hot and Tf-Idf method in data processing phase. Then, these three models were trained and evaluated with the converted dataset. Eventually, the results were analysed and compared with consideration of the models' properties.

2 Related Work

These two papers are about the source and data collection of the movie classification dataset which has been used in this report. These authors mentioned about the reason they collected these data. Plus, they provided brief understanding of how they process and arrange the dataset. [1][2]

~~The authors of~~ this paper [3] also took three machine learning models such as Naïve Bayes, Decision Tree, and Multi-Layer Perceptron into consideration. They compared the performance of these three models on the keyphrases experiment. And they concluded that Multi-Layer Perceptron model outperforms the other two models.

3 Materials

3.1 Dataset

The original datasets of this movie classification test are "tsv" files including train_features, train_labels, valid_features, valid_labels, NEW_test_features. The files with filename "train" and "valid" have both features and labels which can be used to train and validate the models. The file with filename "test" which does not provide correct labels is for the Kaggle competition.

There are three major categories of features inside this dataset including metadata features, visual features, and audio features. Metadata features contain vocabulary type data such as "title", "year of release", "tag", etc.. On the other hand, both visual features and audio features are all in numerical type. In files with filename "labels", each movie is labelled with its genre. Overall, there are 18 genres.

3.2 Data Processing

As mentioned in section 3.1, the metadata features from the original dataset contains vocabulary data which cannot be used by the selected machine learning models especially the feature "tag". For example, the feature "tag" in the original dataset might be "tag1,tag2,…". Thus, the data needs to be converted into usable type before fitting into the models. Two methods such as One-Hot and Tf-Idf has been utilized in this report.

3.2.1 One-Hot

This method will first go through one feature of the whole dataframe and store all different tags or words in a list(see Image 1). Then, for each tag or word inside the list, it will go through every instances in the dataframe and check

whether this tag or word exists in this instance. If yes, the value is 1. Otherwise, the value is 0. After a tag or word has gone through every instances, a new feature named by this tag or word is added to the dataframe. This new feature contains only binary values which indicates whether this feature exists in this instance or not(see Image 2). But the main disadvantage of One-Hot is that every features have the same weight 0 or 1. This might not be comprehensive enough and causes some biases.

['steven_spielberg', 'james_bond', 'matt_damon', 'oscar', 'anime']

Image 1: A list with different tags

steven_spielberg	james_bond	matt_damon	oscar	anime
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Image 2: New features named by the tags with binary value

3.2.2 Tf-Idf

Tf-Idf [4] can overcome the shortcoming of One-Hot. This method will first calculate the term frequency(Tf)(see Formula 1) which can be obtained by using CountVectorizer. The tag might be meaningful if the frequency is high. Second, this method will calculate the inverse document frequency(Idf)(see Formula 2) which represents the frequency of a tag or word appears within instances. A tag is not meaningful at all if it appears in most of the instances. Finally, obtain the Tf-Idf meighted value by multiplying Tf and Idf. Using TfidfTransformer we can obtain a metrix of Tf-Idf weighted value. Then, add it to the dataframe column by column.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Formula 1: Term Frequency

$$idf_{i,j} = \log \frac{|D|}{1 + |D_{t_i}|}$$

Formula 2: Inverse Document Frequency

4 Methods

4.1 Model Overview & Selection

Review some widely used machine learning models' concept and principle. Then, select suitable machine learning models to research and analyze in this report based on each machine leaning model's algorithm and property.

4.1.1 Naïve Bayes

Naïve Bayes is one of the most basic and straight forward model in machine learning. Naïve Bayes is a conditional probability model which uses Bayes' theorem(see Formula 3) to calculate every possible situations' probability. Then, return the predicted class label with maximun likelihood(see Formula 4). [5]

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

Formula 3: Bayes' Theorem

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}}\ p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)$$

Formula 4: NB-Classifier's Final Formula

Due to the fact that Naïve Bayes model only contains simple probability mathematics and fundamental concept. Plus, it often performs surprisingly well. This model was selected as the baseline model of this report.

4.1.2 Decision Tree

The second machine learning modelwhich has been selected is Decision Tree. It has over 1760 thousands relevant papers been published which means it is a popular model in research field.

The idea of Decision Tree [6] is human friendly. To be more specific, the concept of this model is similar to if-else condition and is easy for human to understand. In the tree, each node represents a feature and each feature value becomes a branch of the node. From top to bottom layer, it counts the number of each possible class label and assign a class using the most common class of the subset in each leaf. Plus, the decision of which feature should be in the upper layer depends on Information Gain, Gain Ratio, or Gini. This is one of the key factor of Decision Tree. [7][8][9]

4.1.3 Multi-Layer Perceptron(MLP)

Multi-Layer Perceptron model [10][11] is more powerful than Naïve Bayes and Decision Tree. However, it's more complicated as well.

Multi-Layer Perceptron contains three layer categories such as input layer, output layer, and hidden layer. In input layer, it takes features in dataset as input. On the other hand, output layer has

one unit per possible output which is the predicted class label. Finally, the most important layer category is hidden layer. In MLP, engineer might set up several hidden layers. Each hidden layer takes some parameters of features into consideraiton and passes it down to the next hidden layer. As a result, the more and more relevant or critical features will be passed to next layer and be utilized to predict the class label.

4.2 Model Design & implementation

4.2.1 Naïve Bayes Model

Gaussian Naïve Bayes were applied in this report. This is because the visual features and audio features are all continuous data. And the author assumes that the data satisfies Gaussian distribution. The data in metadata features also satisfies Gaussian distribution after applying Tf-Idf.

Due to the fact that Naïve Bayes classifier makes predictions according to the conditional probability. The predicted result might be affected by the features' weight. Thus, the predicted result using Tf-Idf should outperform the one using One-Hot.

4.2.2 Decision Tree Model

It's important to place the right feature in the upper layer of the decision tree. This is because features appear in the upper layer will be identified first. To be more specific, if the decision tree could identify the feature with the most meaningful information first, the performance of this classifier will become better.

In sklearn library, we could apply either Information Gain(Entropy) or Gini Impurity to our decision tree classifier [12]. This report applies both IG and Gini, and includes the results of these two methods.

4.2.3 Multi-Layer Perceptron Model

In this report, the author applied "adam" as solver in the Multi-Layer Perceptron classifier. This is because the provided dataset has thoudsands of instances and solver "adam" is suitable for large set of data.

Designers can easily adjust the learning rate of the MLP classifier using sklearn library. Learning rate is critical for classifier because it decides how fast our model will modify or learn. In order to find the optimal learning rate, we should find the zero point of the derivative of this model's loss function(see Figure 1). The extreme point of the learning rate-accuracy figure has learning rate at about 0.001. Thus, the author assigned the learning rate to 0.001.

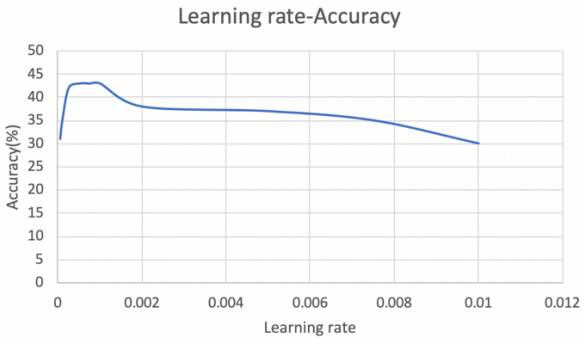

Figure 1: Learning rate-Accuracy

5 Result &Evaluation

5.1 Naïve Bayes Model

	precision	recall	f1-score	support
Action	0.02	0.17	0.04	6
Adventure	0.00	0.00	0.00	2
Animation	0.03	0.67	0.06	3
Children	0.06	0.67	0.11	3
Comedy	0.50	0.03	0.05	38
Crime	0.00	0.00	0.00	5
Documentary	0.29	0.56	0.38	18
Drama	0.44	0.09	0.15	43
Fantasy	0.29	0.11	0.16	18
Film_Noir	0.13	0.50	0.21	4
Horror	0.15	0.25	0.19	8
Musical	0.09	0.10	0.10	10
Mystery	0.33	0.06	0.10	18
Romance	0.50	0.02	0.04	51
Sci_Fi	0.70	0.44	0.54	16
Thriller	0.83	0.18	0.29	28
War	0.43	0.14	0.21	21
Western	0.00	0.00	0.00	7
accuracy			0.15	299
macro avg	0.27	0.22	0.15	299
weighted avg	0.42	0.15	0.16	299

Figure 2: Classification report(Tf-Idf)

	precision	recall	f1-score	support
Action	0.00	0.00	0.00	6
Adventure	0.00	0.00	0.00	2
Animation	0.03	0.67	0.06	3
Children	0.07	1.00	0.13	3
Comedy	0.00	0.00	0.00	38
Crime	0.00	0.00	0.00	5
Documentary	0.22	0.56	0.32	18
Drama	0.67	0.05	0.09	43
Fantasy	0.44	0.22	0.30	18
Film_Noir	0.06	0.25	0.10	4
Horror	0.11	0.12	0.12	8
Musical	0.14	0.10	0.12	10
Mystery	1.00	0.06	0.11	18
Romance	0.50	0.02	0.04	51
Sci_Fi	0.56	0.31	0.40	16
Thriller	1.00	0.04	0.07	28
War	0.20	0.05	0.08	21
Western	0.00	0.00	0.00	7
accuracy			0.11	299
macro avg	0.28	0.19	0.11	299
weighted avg	0.43	0.11	0.11	299

Figure 3: Classification report(One-Hot)

5.2 Decision Tree Model

	precision	recall	f1-score	support
Action	0.00	0.00	0.00	6
Adventure	0.17	0.50	0.25	2
Animation	0.00	0.00	0.00	3
Children	0.00	0.00	0.00	3
Comedy	0.25	0.21	0.23	38
Crime	0.08	0.20	0.12	5
Documentary	0.54	0.39	0.45	18
Drama	0.27	0.33	0.29	43
Fantasy	0.23	0.17	0.19	18
Film_Noir	0.50	0.25	0.33	4
Horror	0.00	0.00	0.00	8
Musical	0.00	0.00	0.00	10
Mystery	0.00	0.00	0.00	18
Romance	0.26	0.24	0.24	51
Sci_Fi	0.26	0.38	0.31	16
Thriller	0.21	0.25	0.23	28
War	0.55	0.29	0.37	21
Western	0.00	0.00	0.00	7
accuracy			0.22	299
macro avg	0.18	0.18	0.17	299
weighted avg	0.24	0.22	0.22	299

Figure 4: Classification report
(Tf-Idf & Gini)

	precision	recall	f1-score	support
Action	0.00	0.00	0.00	6
Adventure	0.00	0.00	0.00	2
Animation	0.17	0.33	0.22	3
Children	0.00	0.00	0.00	3
Comedy	0.33	0.24	0.28	38
Crime	0.10	0.20	0.13	5
Documentary	0.36	0.22	0.28	18
Drama	0.25	0.19	0.21	43
Fantasy	0.42	0.28	0.33	18
Film_Noir	0.00	0.00	0.00	4
Horror	0.12	0.12	0.12	8
Musical	0.10	0.10	0.10	10
Mystery	0.06	0.06	0.06	18
Romance	0.25	0.27	0.26	51
Sci_Fi	0.35	0.69	0.47	16
Thriller	0.21	0.32	0.26	28
War	0.44	0.19	0.27	21
Western	0.25	0.14	0.18	7
accuracy			0.23	299
macro avg	0.19	0.19	0.18	299
weighted avg	0.26	0.23	0.23	299

Figure 5: Classification report(Tf-Idf & Entropy)

	precision	recall	f1-score	support
Action	0.00	0.00	0.00	6
Adventure	0.00	0.00	0.00	2
Animation	0.00	0.00	0.00	3
Children	0.00	0.00	0.00	3
Comedy	0.30	0.29	0.29	38
Crime	0.18	0.40	0.25	5
Documentary	0.46	0.33	0.39	18
Drama	0.22	0.21	0.21	43
Fantasy	0.17	0.17	0.17	18
Film_Noir	0.00	0.00	0.00	4
Horror	0.10	0.12	0.11	8
Musical	0.00	0.00	0.00	10
Mystery	0.12	0.11	0.12	18
Romance	0.22	0.20	0.21	51
Sci_Fi	0.33	0.56	0.42	16
Thriller	0.26	0.32	0.29	28
War	0.43	0.29	0.34	21
Western	0.00	0.00	0.00	7
accuracy			0.23	299
macro avg	0.16	0.17	0.16	299
weighted avg	0.23	0.23	0.22	299

Figure 6: Classification report(One-Hot & Gini)

	precision	recall	f1-score	support
Action	0.00	0.00	0.00	6
Adventure	0.00	0.00	0.00	2
Animation	0.00	0.00	0.00	3
Children	0.00	0.00	0.00	3
Comedy	0.27	0.29	0.28	38
Crime	0.00	0.00	0.00	5
Documentary	0.31	0.22	0.26	18
Drama	0.27	0.21	0.24	43
Fantasy	0.21	0.22	0.22	18
Film_Noir	0.00	0.00	0.00	4
Horror	0.08	0.12	0.10	8
Musical	0.09	0.10	0.10	10
Mystery	0.06	0.06	0.06	18
Romance	0.21	0.14	0.16	51
Sci_Fi	0.26	0.56	0.35	16
Thriller	0.15	0.21	0.18	28
War	0.19	0.14	0.16	21
Western	0.00	0.00	0.00	7
accuracy			0.19	299
macro avg	0.12	0.13	0.12	299
weighted avg	0.19	0.19	0.18	299

Figure 7: Classification report(One-Hot & Entropy)

5.3 Multi-Layer Perceptron Model

	precision	recall	f1-score	support
Action	0.00	0.00	0.00	6
Adventure	0.00	0.00	0.00	2
Animation	1.00	0.33	0.50	3
Children	0.50	0.33	0.40	3
Comedy	0.41	0.39	0.40	38
Crime	0.33	0.20	0.25	5
Documentary	0.65	0.61	0.63	18
Drama	0.49	0.51	0.50	43
Fantasy	0.35	0.33	0.34	18
Film_Noir	0.00	0.00	0.00	4
Horror	0.36	0.50	0.42	8
Musical	0.20	0.10	0.13	10
Mystery	0.57	0.22	0.32	18
Romance	0.32	0.61	0.42	51
Sci_Fi	0.67	0.50	0.57	16
Thriller	0.45	0.54	0.49	28
War	0.67	0.29	0.40	21
Western	0.00	0.00	0.00	7
accuracy			0.42	299
macro avg	0.39	0.30	0.32	299
weighted avg	0.43	0.42	0.41	299

Figure 8: Classification report
(Tf-Idf & Learning rate=0.001 & solver=adam)

	precision	recall	f1-score	support
Action	1.00	0.17	0.29	6
Adventure	0.00	0.00	0.00	2
Animation	0.50	0.33	0.40	3
Children	0.50	0.33	0.40	3
Comedy	0.41	0.61	0.49	38
Crime	0.00	0.00	0.00	5
Documentary	0.46	0.72	0.57	18
Drama	0.54	0.44	0.49	43
Fantasy	0.71	0.28	0.40	18
Film_Noir	0.00	0.00	0.00	4
Horror	0.38	0.75	0.50	8
Musical	0.25	0.10	0.14	10
Mystery	0.67	0.11	0.19	18
Romance	0.34	0.47	0.40	51
Sci_Fi	0.62	0.62	0.62	16
Thriller	0.34	0.61	0.44	28
War	0.71	0.24	0.36	21
Western	0.00	0.00	0.00	7
accuracy			0.43	299
macro avg	0.41	0.32	0.32	299
weighted avg	0.46	0.43	0.40	299

Figure 9: Classification report
(One-Hot & Learning rate=0.001 & solver=adam)

6 Discussion & Error Analysis

		One-Hot				Tf-Idf			
		NB	DT(Gini)	DT(Entropy)	MLP	NB	DT(Gini)	DT(Entropy)	MLP
Accuracy		0.11	*0.23	*0.19	0.43	0.15	0.22	0.23	0.42
Macro Avg.	precision	0.28	0.16	0.12	0.41	0.27	0.18	0.19	0.39
	recall	0.19	0.17	0.13	0.32	0.22	0.18	0.19	0.3
	f1-score	0.11	0.16	0.12	0.32	0.15	0.17	0.18	0.32
Weighted Avg.	precision	0.43	0.23	0.19	0.46	0.42	0.24	0.26	0.43
	recall	0.11	0.23	0.19	0.43	0.15	0.22	0.23	0.42
	f1-score	0.11	0.22	0.18	0.4	0.16	0.22	0.23	0.41

Table 1: Overall Comparision of MLP, DT, NB

6.1 Overall Comparision

According to the analysis from classification reports(see Table 1), MLP siginificantly outperforms other two models. Take a deep look of it's algorithm properties we can discover that with the help of multiple layers, MLP is comfortable with non-linear problems. On the contrary, Naïve Bayes is a linear classifier. Thus, when it comes to movie classification which is non-linear, Naïve Bayes' performance becomes quite poor.

Comparing the label distribution charts of these three models(see Figure 10, 11, 12). The orange line in the chart represents the correct predicted label. Thus, the model who's label distribution is more concentrated to the orange line, the better performance it has. As we can observe, MLP's distribution is the most concentrated one. On the other hand, Naïve Bayes has the most wide spread one.

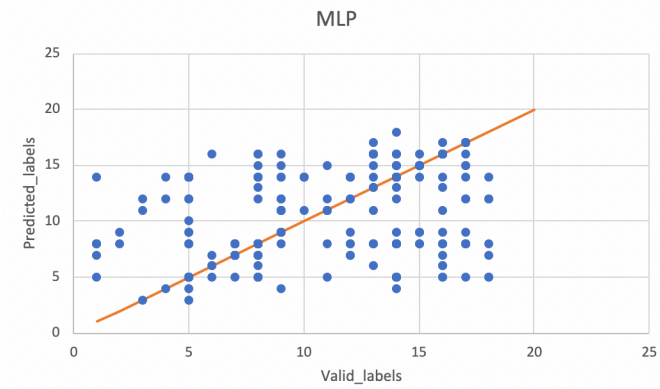

Figure 10: Predicted-Valid label distribution of MLP

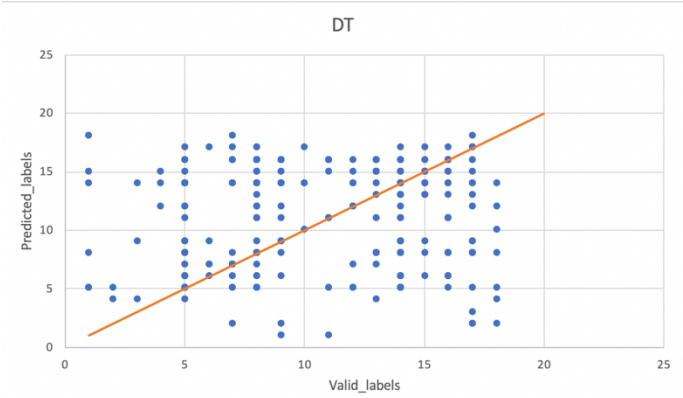

Figure 11: redicted-Valid label distribution of DT

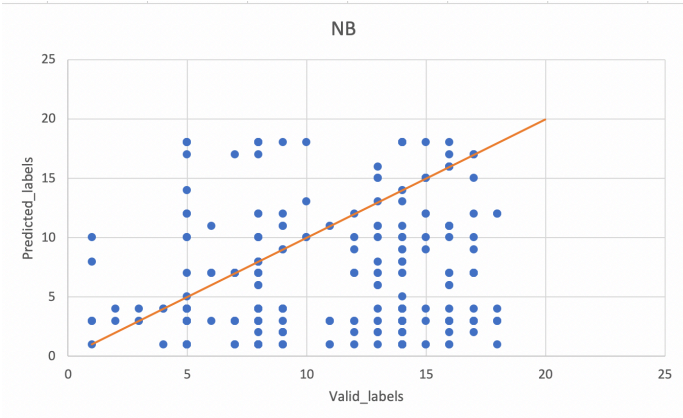

Figure 12: redicted-Valid label distribution of NB

6.2 Naïve Bayes' Shortcoming

There major reason that lead to NB's poor performance is the assumption of every features and instances are independent to each others. For example, the visual and audio features in this report might influence other features. The second reason caused even worst performance. By using One-Hot to process the data,

it only contains binary values but this report used Guassian NB. Thus, the binary value will be too rigid for Gaussian distribution.

6.2 Analysis of Decision Tree

In this report, Information Gain(Entropy) and Gini Impurity has been applied to DT. However, Information Gain has a shortcoming that for an feature with a large number of values the subsets are more likely to be pure which might result in overfitting.

The feature values in visual, audio, and converted metadata are all continuous data which means an feature might has a large number of values. Thus, the performance of DT using entropy method has been influenced(see Table 1's data with * mark).

7 Reference

[1] Deldjoo, Yashar and Constantin, Mihai Gabriel and Schedl, Markus and Ionescu, Bogdan and Cremonesi, Paolo. MMTF-14K: A Multifaceted Movie Trailer Feature Dataset for Recommenda- tion and Retrieval. Proceedings of the 9th ACM Multimedia Systems Conference, MMSys 2018, Amsterdam, The Netherlands, June 12-15, 2018

[2] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015)

[3] Kamal Sarkar*, Mita Nasipuri* and Suranjan Ghose* December 2012 Machine Learning Based Keyphrase Extraction: Comparing Decision Trees, Naïve Bayes, and Artificial Neural Networks

[4] Wikipedia contributors. (2020, January 9). TF. In *Wikipedia, The Free Encyclopedia*. Retrieved 04:30, May 22, 2020, from https://en.wikipedia.org/w/index.php?title=TF&oldid=934940947

[5] Wikipedia contributors. (2020, May 20). Naive Bayes classifier. In *Wikipedia, The Free Encyclopedia*. Retrieved 04:31, May 22, 2020, from https://en.wikipedia.org/w/index.php?title=Naive_Bayes_classifier&oldid=957801494

[6] Wikipedia contributors. (2020, May 20). Decision tree. In *Wikipedia, The Free Encyclopedia*. Retrieved 04:32, May 22, 2020, from https://en.wikipedia.org/w/index.php?title=Decision_tree&oldid=957678776

[7] Mitchell, Tom (1997). Machine Learning. Chapter 3: Decision Tree Learning.

[8] Tan et al (2006) Introduction to Data Mining. Section 4.3, pp 150-171.

[9] Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81–106

[10]Jacob Eisenstein (2019). Natural Language Processing. MIT Press. Chapters 3 (intro), 3.1, 3.2. https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf

[11]Dan Jurafsky and James H. Martin. Speech and Language Processing. Chapter 7.2, 7.3. Online DraftV3.0.https://web.stanford.edu/~jurafsky/slp3/

[12] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.